

## 分散メモリ型と共有メモリ型マルチプロセッサによる 並列配線処理の性能評価†

佐野 雅彦<sup>††</sup> 高橋 義造<sup>†††</sup>

われわれの研究室では、各種アプリケーションに対して有効な並列処理アルゴリズムの研究を行っているが、その一貫として配線問題の並列処理を取り上げた。これまでの研究結果より、1層の配線問題に対してプロセッサ競合方式とネット割り当て法を用いた処理方式の有効性を確認したが、1層の配線問題では現実性に欠けるので、今回は同じ並列処理方式を2層配線問題に適用する研究を行い、その結果、この場合にも本処理方式の有効性を確認することができた。さらにこの方式を分散メモリ型と共有メモリ型の両方式のマルチプロセッサに適用した場合の性能の比較を行った。その結果、本方式は共有メモリ型マルチプロセッサに対してより有効であることが確認された。

### 1. はじめに

われわれは、実社会で要求される各種アプリケーションにマルチプロセッサを適用するために、個々の問題に適した処理アルゴリズムの研究を行っている。一般に、これらのアプリケーションに従来から使用されている逐次処理方式をそのまま単純に並列化することは難しく、各アプリケーションごとにそれに適した処理アルゴリズムを用いる必要がある。

マルチプロセッサには二つの方式がある。一つは分散メモリ型マルチプロセッサ（マルチコンピュータとも呼ばれる）であり、大規模なマルチプロセッサを構成することが可能である。他の一つは共有メモリ型マルチプロセッサであり、大規模な構成は困難であるが、プロセッサ間通信が高速であるという特長を持つ。これらのマルチプロセッサに対する効率の良い実装方法はアーキテクチャに依存し、アーキテクチャごとに異なる処理方式を開発しなければならないという問題がある。そこで、われわれの研究室ではいくつかの実際のアプリケーションを取り上げ、それに対する並列処理アルゴリズムとアーキテクチャへの依存性の低い実装方式を開発するための研究を行っているが、その一つに配線問題の並列処理がある。これまでの研究結果よりプロセッサ競合方式とネット割り当て法の複合処理方式の有効性が確認されている<sup>1),2)</sup>。しかし、

そこで取り扱ったのは1層の配線問題であったため、実データを用いた既存の配線処理プログラムとの性能比較を行うには不十分であった。

そこで、今回は2層配線問題に対してプロセッサ競合方式とネット割り当て法の複合並列処理方式を適用した場合の有効性を確認し、分散メモリ型と共有メモリ型の異なるマルチプロセッサ方式において、本方式の性能を比較することを目的とした研究を行った。その結果、本方式がこの問題に対して有効であり、両方式のマルチプロセッサに対しても有効であるが、特に共有メモリ方式のマルチプロセッサにおいて優れていることが確認できた。

### 2. 配線問題の並列処理

#### 2.1 問題の設定

本研究で取り扱う配線問題はプリント配線基板の配線を対象とするが、高度な配線アルゴリズムの研究が目的ではなく、並列処理アルゴリズムの研究を目的としたため、以下のような単純な問題を設定した。

- 1) 配線領域は2層とする。
- 2) 各層の配線領域のグリッドサイズは  $256 \times 256$  とする。
- 3) 各ピンはグリッド上に配置され、グリッド間は配線しない。
- 4) 各層における配線方向は、縦または横方向を原則とする。
- 5) ネットの始点と終点の座標の組をネットデータとして与え、この間の配線経路を求める。

基本的な配線アルゴリズムには、迷路法、線分探索法などがあるが、本研究では線分探索法を用いた。線分探索法は与えられた2点間を接続するために各々の

† Performance Evaluation of a Parallel Wire Routing Algorithm on Distributed Memory and Shared Memory Multiprocessors by MASAHICO SANO (Department of Information Science and Intelligent Systems, Faculty of Engineering, Tokushima University) and YOSHIZO TAKAHASHI (Department of Information Science and Intelligent Systems, Faculty of Engineering, Tokushima University).

†† 徳島大学大学院工学研究科情報工学専攻

††† 徳島大学工学部知能情報工学科

端点より探索線を交互に順次延ばし、相互の探索線が交差するまで繰り返し行う配線方法であり、迷路法に比べて処理時間が短いことが特長とされている<sup>3)</sup>。

## 2.2 本研究で使用した並列処理方法

これまでにいくつかの並列配線処理方法が提案されているが<sup>7)-9)</sup>、われわれはこれまでの研究結果より<sup>1),2)</sup>、処理のスケジューリング方式として「**プロセッサ競合方式**」を、処理の分割方法に「**ネット割り当て法**」を用いた。**プロセッサ競合方式**は「早い者勝ち」で解決する方式で、処理モデルが簡単であり、プロセッサ要素（以下 PE と呼ぶ）間の競合の調停が単純であるという特長を持つ。**ネット割り当て法**は 1本のネットを処理単位として、各 PE に割り当てる方式である。この方式は問題の分割が簡単であり、高い並列性が実現できるという特長がある。

次の 2.3 節、2.4 節では本研究で使用した並列処理方式の**基本計算モデル**について述べる。

## 2.3 計算モデル

配線処理を並列化する場合、各 PE は同じ配線領域上で作業するので、すべての PE はこの領域を参照する必要がある。本研究ではネット割り当て法とプロセッサ競合方式を組み合わせた計算モデルを用いる。このモデルは図 1 に示すように、1 台のマスタープロセッサ（以下マスター）と多数のスレーブプロセッサ（以下スレーブ）、およびデータベースから構成される。マスターとスレーブ間は適当な方法で結合され、相互通信が可能である。データベースには配線処理に必要なデータや処理結果が格納されており、スレーブの要求に応じて提供される。また、計算機内部ではデータベースの物理的な位置に制約はないものとする。

## 2.4 処理アルゴリズム

マスター/スレーブ方式の基本的な処理手順は、次のようである。マスターは各スレーブに異なる 1 本のネッ

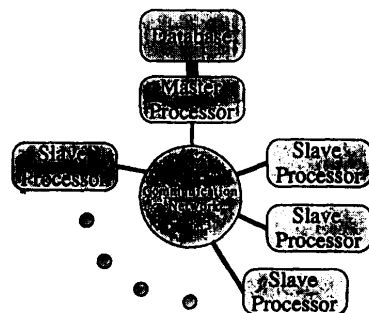


図 1 競合プロセッサによる計算モデル  
Fig. 1 Processing model of parallel computing with competing processors.

トを割り当てる。各スレーブは割り当てられたネットを独立に配線処理し、結果をマスターに送る。配線結果を受け取ったマスターはその配線結果が既配線ネットと衝突していなければデータベースに登録し、衝突すれば再配線処理をスレーブに指示する。ここで、マスターが配線結果を受け取る時の順番は先着順である。つまり、2本の配線結果が衝突している場合、マスターは先に送られた配線結果を受け付け、後から送られてきた配線結果を再配線させる。各処理ステップの詳細を次に示す。

(1) **配線領域割り当て** 各スレーブは与えられる 1本のネットデータを受け取るごとに処理を行うが、各スレーブの担当する配線領域の範囲が問題になる。つまり、配線処理されるネットは配線領域全体に及ぶため、配線領域が複数に分割されていると、配線処理のために隣接する配線領域のスレーブと通信を行う必要が生じる。したがって、ネット割り当て法を用いる場合には、配線領域を分割すると、隣接するスレーブ間の通信量が增大するので処理効率が下がる。

われわれは通信量の抑制と処理方式の単純化に考慮して、配線領域全体をすべてのスレーブに割り当てる方式を採用した。これにより、各スレーブは割り当てられたネットを独立に配線することができるようになる。このようにしても、配線領域の参照は個々のスレーブでは局所的であり、スレーブ群全体では領域全体に分散するので、各スレーブの配線領域の参照が局所的に集中する確率は十分低いものと考えられる。

(2) **共通領域の参照と更新** 各スレーブが共通の配線領域上で配線処理を行うので、各時点におけるすべての配線結果を保存するための共通領域を図 1 に示すようにデータベース内に設ける。この共通領域はマスターが適時更新するものとする。一方、各スレーブは共通領域を自由に参照しながら配線処理を行うが、共通領域の更新はできないものとする。

この処理方式では、一台のスレーブが配線中に他のスレーブの配線結果が新たに加えられることがあるので、各スレーブの参照結果に対しての一貫性が保証されない。このため、各スレーブの配線結果に矛盾が生じる恐れがある。この矛盾は(4)で述べる再配線処理を行うことで解決する。すべての共有データに対して一貫性を保証すれば常に正しい結果が得られるが、それによる処理効率の低下が起きるので、このような方式を用いた。

(3) **配線処理** 各スレーブはマスターから与えられ

たネットデータを、共通領域を参照しながら配線処理を行う。配線処理中に発生する一時的なデータはすべてローカルに処理される。

(4) **再配線処理** 共通領域が参照自由であるため、各プロセッサの配線結果の間に矛盾が生じることがある。この矛盾を解消するために再配線処理を行う。

各スレーブは配線処理を終えるとその配線結果をマスタへ送り、マスタは各スレーブから送られてきた配線結果を共通領域と比較して、既配線結果と衝突しているかどうかを検証する。検証の結果、衝突していなければ配線結果を共通領域に登録してデータベースを更新し、衝突していればその配線結果を処理したスレーブに再配線処理を行わせる。このようにして各スレーブの処理結果の間に矛盾があっても再配線により矛盾が解消されるが、再配線回数が過大になると処理能力が低下することになる。

これらの処理の流れを図2を用いて説明すると次のようになる。

- ①個々のネットをスレーブに割り当てる。
- ②スレーブは共通領域を参照しながら配線処理を行う。
- ③処理結果をマスタに送る。
- ④スレーブから受け取った処理結果をマスタは共通領域と比較して検証を行う。その結果、他の既配線結果と衝突していなければ共通領域を更新する。そうでなければ再配線処理させる。

これらの処理はマスタとスレーブによって分担され、マスタにおける処理項目は次の3点である。

(1) **配線処理結果の検証と登録** 各スレーブでの配線結果が正しいかどうかを検証を行う。検証にはスレーブの配線結果をトレースして他の処理済みの結果と衝突していないかどうかを調べる方法を用いる。検

証の結果、衝突がなければ正しい配線結果として共通領域に登録する。

(2) **複数のスレーブの競合の裁定** 各スレーブは1本のネットの配線処理の終了後、マスタへ配線結果を送り検証を依頼する。しかし、他のスレーブでも同じ動作をするため競合が発生する。そこで、この競合を「早い者勝ち」順に裁定する。

(3) **データベースの管理** 共通領域と各スレーブで配線処理されるネットデータ、スレーブが処理したすべての配線結果はデータベース内部に格納され、マスタが管理する。

スレーブの処理項目は次の2点である。

- (1) マスタより配線すべきネットデータを1本ずつ受け取り共通領域を参照しながら配線処理を行う。
- (2) 配線処理終了後、配線結果をマスタに送る。

### 3. 分散メモリ型マルチプロセッサによる並列配線処理

#### 3.1 処理方式

分散メモリ型で用いた処理方式は、2.3節、2.4節で述べた基本計算モデルに以下の変更を加えたものである。

(1) **共通領域と参照方式の変更** 基本計算モデルでは共通領域は全PEから参照可能であると考えたが、分散メモリ型マルチプロセッサには共有メモリが存在しないため、プロセッサ間通信を行う必要がある。そこで、各スレーブの配線結果に矛盾を許すという特徴を利用して、共通領域を配線処理の開始時にコピーする方式を用いることにし、これによって生じる矛盾は再配線処理により解消する。再配線処理の回数が高くなると処理効率が低下するが、以前の研究結果<sup>1),2)</sup>より、再配線処理の発生割合は全配線処理の高々10%程度であることが確認されているため、分散メモリ型においては十分有効な方式であると考えられる。

(2) **共通領域のコピー方式** コピーには部分コピーと全体の2方式が考えられる。部分コピー方式は、図3(a)に示すようにスレーブの配線処理の開始時に、マスタがスレーブに割り当てるネットから配線範囲(図中の塗り潰された領域)を推定し、その領域だけのコピーを行う方式である。もしもスレーブの配線処理中に処理範囲がコピーされた領域をはみ出す場合があると、はみ出した部分の共通領域のコピーをマスタからもらって配線処理を継続する。この方式で

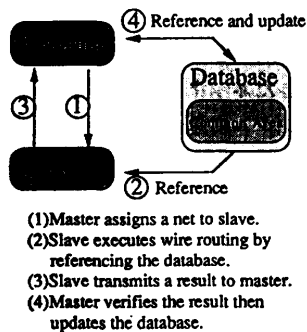


図2 処理の流れ

Fig. 2 Sequence of processing.

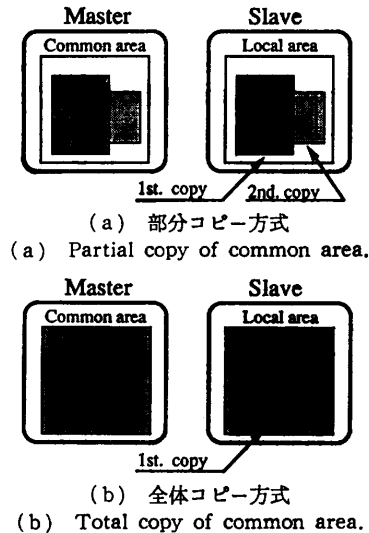


図 3 共通領域のコピー方式

Fig. 3 Two ways of copying common area.

は、配線処理開始時のデータ転送量はかなり低減できるが、通信処理の増加のため、処理効率が低下することが考えられる。

一方、全体コピー方式は図 3 (b) に示すように共通領域全体を配線処理開始時にコピーする方式である。部分コピー方式に比べ簡単であるが、共通領域のサイズが大きいと転送時間が長くなることが欠点になる。しかし転送時間は一定であり、コピー回数も処理の開始時に行う 1 回だけで済むという利点もある。本研究では取り扱う配線問題の規模を考慮して後者の全体コピー方式を用いた。

分散メモリ型マルチプロセッサにおける処理は図 4 に示すとおり、次のような手順で行われる。

- ① マスタは分割したネットと共通領域のコピーをスレーブに転送する。
- ② スレーブはマスタから配線処理するネットと共通領域のコピーを受け取り、このコピーを参照しながら独立に配線処理を行う。
- ③ スレーブは配線処理結果をマスタに送る。
- ④ マスタはスレーブから受け取った配線処理結果を共通領域上で検証し、この領域の更新を行う。

この処理方式を評価するための分散メモリ型マルチプロセッサとして Coral 68 K を用いた。Coral 68 K は 1987 年に、われわれの研究室で開発された分散メモリ型の实用規模の 2 進木結合マルチプロセッサであり、MPU に MC 68000 (10 MHz) を用いた 63 台の PE を図 5 のように 2 進木状に結合したものである。PE 間の通信には DMA 通信を用いており 2MB/

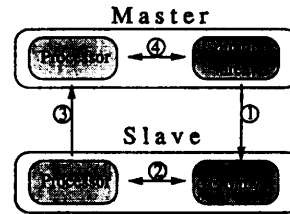


図 4 分散メモリ型における処理の流れ

Fig. 4 Sequence of processing on the distributed memory multiprocessor.

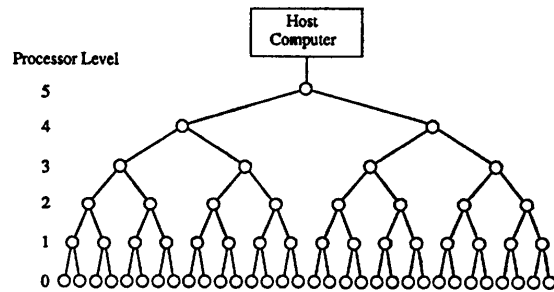


図 5 Coral 68 K の構成

Fig. 5 Organization of Coral 68 K.

秒の転送速度を持つ。各 PE は 512KB のメモリを持ち、演算能力は全体で 40 MIPS, 0.3 MFLOPS である<sup>4),5)</sup>。なお、今回の配線処理プログラムでは浮動小数演算は使用していない。

Coral 68 K による並列配線処理では、先で述べた分散メモリ型の場合の処理方式を次のように適用した。

- (1) 2 進木のルートの PE をマスタとし、その他の全 PE をスレーブとする。
- (2) データベースをマスタ内部に設け、スレーブによる共通領域の参照は、先に述べた共通領域の全体コピー方式を用いて実現する。
- (3) 2 進木状に結合されているため、マスタから離れたスレーブではマスタと通信するために、他のスレーブによる中継を必要とする。この中継の処理には割り込み処理を用いて行う。

### 3.2 配線処理結果

評価に使用したネットデータとして、 $256 \times 256$  のグリッドからランダムに選んだ始点・終点間のマンハッタン距離の平均が 10, 20 になるようなランダムネットを、ネット数 1,000, 2,000, 3,000 の場合について作成した。図 6 にネット数 3,000 の場合のマンハッタン距離の分布を示す。図 7 は 3,000 本のネットの配線を行った場合で、(a) は速度向上比とプロセッサ利用率、(b) はその時の配線率と再配線率、(c) は 500 ネットごとに集計した配線率と再配線率のグラフであ

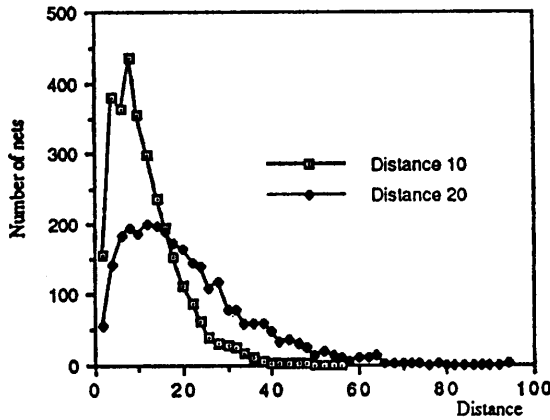


図 6 ランダムネットのマンハッタン距離の分布  
Fig. 6 A distribution of manhattan of random net.

る。配線率は平均距離 10 の場合 98.3%，平均距離 20 の場合 83.5% となった。

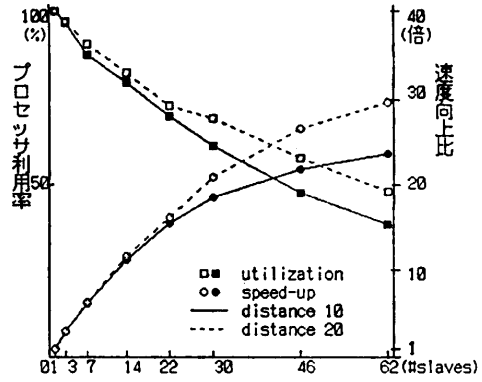
この結果から、スレーブの台数の増加に従って利用率は低下し、スレーブ台数が少ない場合においても台数が多い場合と同様な利用率の低下が見られる。この低下の原因としては先に述べた PE 間通信の増加と、マスタへの負荷集中の二つが考えられるが、前者の PE 間通信によるものが主な原因であると推測できる。なぜなら、今回使用した並列計算機は 2 進木結合であり、通信の中継は PE を用いているため、通信の中継回数は上位の PE ほど多くなる。その結果、台数が多い場合、上位の PE は通信に大部分の時間を費やすことになる。また、マスタへ処理が集中しやすくなるため処理効率も低下する。平均距離の短い場合のほうが処理効率が悪いのは、平均距離が短いと配線時間が短くて済むため、マスタに集中する単位時間当りの処理量が増加するためである。

再配線の割合は 6% 未満であった。よって、共通領域のコピーによる配線結果の矛盾の割合は小さいといえる。この結果から、配線時に共通領域全体のコピーを用いる方法は分散メモリ型に対しては、PE 間の通信時間が増加しプロセッサ利用率の低下を招いているにしても、全体としては有効であることがわかる。

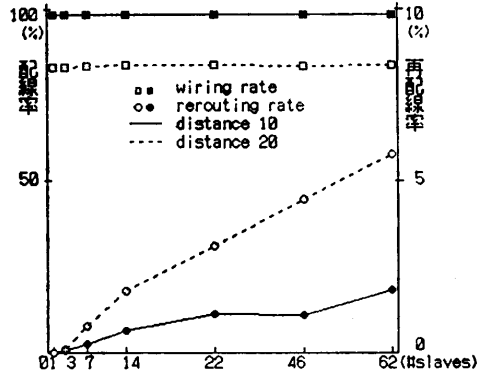
#### 4. 共有メモリ型マルチプロセッサによる並列配線処理

##### 4.1 処理方式

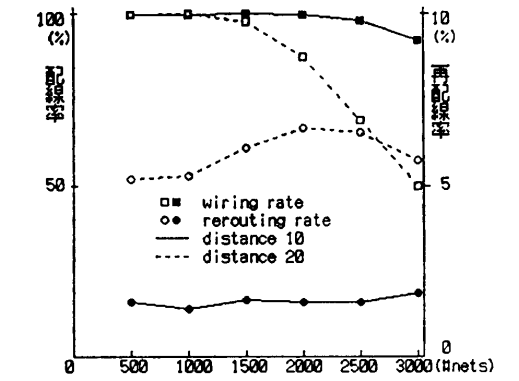
共有メモリ型マルチプロセッサの場合の計算モデルは基本計算モデルと似ているため実現することは容易であるが、分散メモリ型マルチプロセッサの場合の評



(a) プロセッサ利用率と速度向上比  
(a) Processor utilization and speed up rate.



(b) 配線率と再配線率  
(b) Wiring rate and rerouting rate.



(c) 500 ネット単位の配線率と再配線率  
(c) Wiring rate and rerouting rate by 500 nets.

図 7 Coral 68 K の配線処理結果  
Fig. 7 Result of parallel wire routing on Coral 68 K.

価結果から以下の考察を行い、基本計算モデルの方式に多少の変更を加えた。

分散メモリ型での配線結果より、プロセッサ利用率が使用 PE 台数の増加に従い低下することが確認された。この原因は、PE 間通信とマスタへ処理が集中するためであると考えられる。今回使用した並列計算機

は2進木結合であり、上位のPEほど通信が集中し、ボトルネックが生ずる。本方式ではすべてのスレーブがマスタと通信を行うので、マスタに通信が集中しやすくなっている。さらに、通信の集中に比例して検証の回数も増加するのでマスタでの処理量の飽和が問題になる。もし、通信と各PEの処理のバランスが取れているならばさらに良い結果が得られるはずである。(一層の場合では62台のスレーブを用いて、PE利用率85%、速度向上比53.7倍となっている<sup>11,2)</sup>。)

一方、共有メモリ型では共通領域の参照がハードウェア的に可能になるので、共通領域全体のコピーの必要性がない。したがって、分散メモリ型で行っていた共通領域のコピーに消費する通信時間がほとんどなくなり、PEの使用効率が向上すると予想される。そこで、共有メモリ型マルチプロセッサにおける計算モデルと処理方式を次のように変更する。

(1) **マスタPEの削除** マスタPEを取り除き、すべてのPEがマスタの処理とスレーブの処理を行う方式に変更する。この方法では配線終了した各スレーブがマスタとなりマスタの処理を行うことにより、マスタとスレーブ間の通信が省略できるので、処理効率の改善が図られる。

(2) **データベースの更新の排他制御** (1)により、複数のマスタが同時に存在する場合がある。その結果、データベースの内容を常に正しく保つには排他制御が必要になる。そこでデータベースに対して排他制御を行うことにより、複数のマスタが同時に存在してもデータベースの内容を正しく保つようにした。この排他制御はマスタの処理の際にのみ行われ、マスタが更新中の誤ったデータを参照したスレーブの配線結果は検証の段階で排除されるので、スレーブの排他制御は不用である。

(3) **共通領域の参照** 共通領域は共有メモリ内に設け、配線処理中の各スレーブは共通領域の参照のみ行い、一時的な更新は各スレーブのローカル・メモリ内で行う。また、共有メモリ型の特徴を生かすため、分散メモリ型とは異なり、スレーブが必要な時、必要な部分のみ参照することにより無駄を省く。このためのコピー動作は、スレーブの配線処理中に自由に行えるので、共有メモリへのアクセスの競合はあるが、配線処理が他のスレーブに邪魔されることがない。そのためPE間通信の減少が期待される。

(4) **配線結果の検証と共通領域の更新** スレーブとしての配線終了後、マスタの処理を行うPEは共通

領域の更新権を取得し、自分自身の配線結果の検証を行う。その結果が正しければ共通領域を更新し、更新権を放棄する。放棄した時点で、他のPEによるマスタの処理が可能になる。検証結果が不正であれば直ちに更新権を放棄し、他のPEへ権利を譲り、再配線処理を行う。

共有メモリ型における処理は、マスタの処理とスレーブの処理がすべてのPEで行われ、これらの処理はマスタとスレーブの二つのモードに分かれる。

**マスタモード**では、主にスレーブモードで処理した配線結果の検証と共通領域の更新、新しいネットデータの取単、配線結果の保存などの処理を行う。

**スレーブモード**では配線開始時に、マスタモードで取得したネットデータを用いて配線処理を行い、配線終了後、マスタモードに変わり、マスタの処理を行う。

このモデルの処理手順は図8を用いて説明すると次のようになる。(ただし、図中のPEの状態は配線処理を始める前の状態とする。)

①マスタモードで得られたネットデータを共通領域を参照しながら配線処理を行う。

②配線処理終了後、マスタモードへ移行する。

③配線処理した結果より、配線結果の検証を行う。検証の結果正しい結果なら共通領域を新たに更新し、新しいネットデータを得る。そうでなければ、同じデータを用いて再配線処理とする。

④マスタモードからスレーブモードへ移行する。その後、①へもどる。

この並列処理モデルの性能評価には、日本IBM(株)東京基礎研究所で開発された共有メモリ型マルチプロセッサTOP-1を使用させていただいた。これは、図9に示すように、MPUとしてi80386を用いた10台のPEが共有バスによって接続されている。この共有バスは64ビット、2重化インタリーブ方式によって、85MB/秒の実効転送能力を持つ。各PEは128KBのスヌープ・キャッシュを持ち、全体で、平均約30MIPS、最大30MFLOPSの処理能力を持つ<sup>6)</sup>。10台のPEのうち、システムの監視に1台、OS(UNIX)

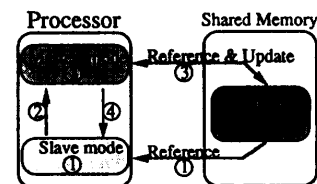


図8 共有メモリ型の場合の処理の流れ  
Fig. 8 Sequence of processing on shared memory multiprocessor.

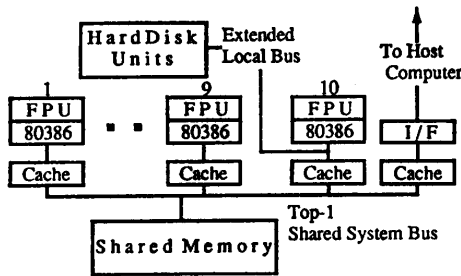


図9 TOP-1の構成  
Fig. 9 Organization of TOP-1.

に1台, 他のプロセスのために1台が割り当てられているので, 実際に使用できる PE は 7~8 台程度である。

4.2 配線処理結果

評価に用いたネットデータは, 図6の分散メモリ型で使用したのと同じものを用いた。図10にネット数3,000本の場合の結果を示す。(a)は速度向上比とプロセッサ利用率, (b)は配線率と再配線率, (c)は500ネットごとに集計した配線率と再配線率のグラフである。配線率は平均距離10の場合98.3%, 平均距離20の場合84.9%となった。

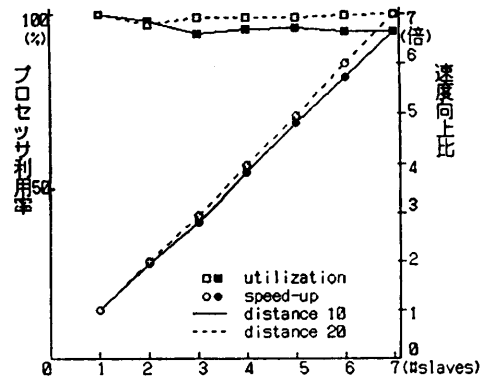
実験結果から90%以上の高い利用率が得られることが分かった。再配線率が1%未満と非常に低く抑えられていることから, 共有メモリの特徴を生かした共通領域の参照方法は共有メモリ型に対して高い有効性をもつと推測される。また, マスタPEを省いたことがある程度の効果を発揮していると思われる。再配線の変化のグラフが上下に変動しているのは, 再配線の発生する時間の不確定性によるものである。

5. 配線処理結果の考察

5.1 両マルチプロセッサ方式に対する性能の比較

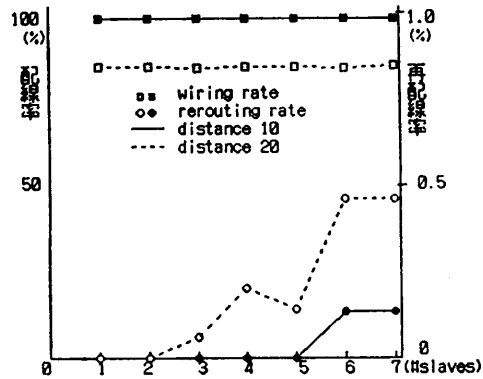
使用した2台のマルチプロセッサの規模が異なるために, 両マルチプロセッサ方式による処理性能を一概に比較することは難しいが, ここでは同じネットデータを同じスレーブ台数で処理した場合の結果を用いて比較を試みる。用いたデータは図6の平均距離20, ネット数3,000のランダムデータで, 先の評価で各々使用したものである。

図11(a)は速度向上比とプロセッサ利用率のグラフである。このグラフから, 両者のプロセッサ方式の結果に大きな差はないが, 共有メモリ型のほうが若干良い結果が出ている。この理由は, 配線処理に必要な部分のみをコピーするTOP-1の参照方式の処理効率が, 配線領域全体をコピーするCoral 68Kのもの



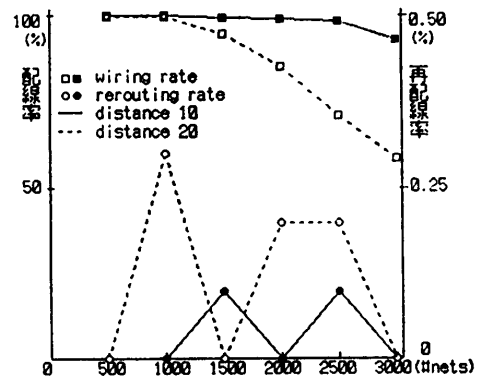
(a) プロセッサ利用率と速度向上比

(a) Processor utilization and speed up rate.



(b) 配線率と再配線率

(b) Wiring rate and rerouting rate.



(c) 500 ネット単位の配線率と再配線率

(c) Wiring rate and rerouting rate by 500 nets.

図10 TOP-1の配線処理結果

Fig. 10 Result of parallel wire routing on TOP-1.

比べて優れているためと考えられる。図11(b)では配線率と再配線率が比較されている。両者の配線率にはほとんど差はないが, 再配線率を見ると, 分散メモリ型のほうが共有メモリ型に比べて約3倍大きい。これは共通領域の参照方法の違いによるものと思われる。すなわち, 共有メモリ型ではスレーブはいつでも共通領域を参照できるため, 最新の情報を使用してい

るので再配線の発生率が低くなっているのである。一方、分散メモリ型では配線開始時にしか共通領域を参照できないため、スレーブの台数が増加するに従い、独立に行われる配線結果が衝突する確率が増加することになる。すなわち、図 11 (b)に見られるようにスレーブの台数にほぼ比例した再配線率になることが推測される。

以上のことから、両プロセッサ方式による処理結果を比較した場合、共有メモリ型のほうが分散メモリ型より有効性が高いということが確認できる。

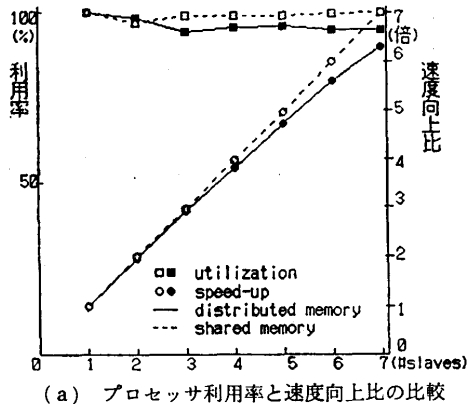
5.2 実データとランダムネットの比較

ランダムネットによる実験だけからこの配線アルゴリズムの評価を下すのは早計であると考え、実データとランダムネットによる配線処理結果の比較を行った。実データとして、イビデン(株)で実際に作成された4層プリント基板のデータを使用させていただ

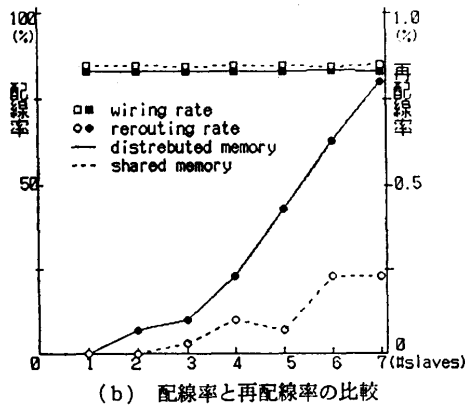
た。この基板はピン間2本の配線規則で処理されており、2,460本のネットが用いられている。評価には分散メモリ型マルチプロセッサ Coral 68 K を使用し、配線層の2層を用いた。配線問題の条件を同じにするために、実データに対応して表1に示すランダムネットを作成した。

これらのネットデータを処理するため、並列配線プログラムを以下のように変更した。変更点は、1) グリッドサイズを実データに合わせて 355×340 に広げる、2) ピン間を2本に拡張する、の2点である。ピン間2本にするということは、ネットの端子位置を3の倍数の格子座標に上にか置かないようにすることに相当する。

このプログラムにより配線処理を行った結果を表2に示す。配線率には大きな差が見られないが再配線率に際だった差が確認される。このため実データの処理時間が大幅に増えている。これは、実データが領域中に局所的に集中して分布しているのに対しランダムネットは一様に分布しているためであると考えられる。このことから配線アルゴリズムの評価に対してランダムデータはかなり有効であるが、データを作成する条件を慎重に決めなければならないことが分かる。条件としては、平均距離、分散、ネット長の分布、配置の分布、各ネット間の相関、等を考慮する必要があるが、今回使用したデータでは最初の3条件は考慮したが後者の2条件は考慮されていない。このことが再配線率の差となって現れているようである。しかしながら実データでは前記の条件を任意に選ぶことができないので、公平なテストデータとして用いるには不十分である。一方、われわれの使用したランダムネットはその生成条件さえ慎重に設定すればアルゴリズムを評価するための公正なテストデータとして使用することができるという利点がある。



(a) プロセッサ利用率と速度向上比の比較  
(a) Comparison of utilization and speed up rate.



(b) 配線率と再配線率の比較  
(b) Comparison of wiring rate and rerouting rate.

図 11 分散メモリ型と共有メモリ型マルチプロセッサの性能比較  
Fig. 11 Performance comparison of distributed and shared memory multiprocessors.

表 1 ランダムネットと実データ  
Table 1 Random net and real data.

	ネット数	マンハッタン平均距離	分散
ランダムネット	2,460	47.6	52.5
実データ	2,460	47.2	44.5

表 2 配線結果  
Table 2 Results of parallel wire-routing.

	配線率	再配線率	処理時間(秒)
ランダムネット	92.9%	13.2%	456
実データ	85.6%	54.2%	1,300



## 6. ま と め

本研究で用いたプロセッサ競合方式とネット割り当て法は2層配線問題に対しても有効であることが再確認された。また、異なるプロセッサ方式に対してはどちらの方式に対しても有効であるが共有メモリ型のほうが有効性が高いことが確認された。しかし、共有メモリ型ではプロセッサ台数を多数使用することは困難であるが分散メモリ型では容易であることを考慮すると、本方式は分散メモリ型の場合にも十分有効な方式であると思われる。さらに、より大規模な問題を解くにはある程度効率が悪くても絶対的な処理能力で優れている分散メモリ型のほうが適していると考えられる。

性能評価に用いるランダムネットは、その作成条件に留意すれば極めて有効なテストデータとして使用できることが確認された。

現在の問題は、並列配線処理の配線率が不十分なことである。実際にイビデン(株)による SUN 3 上で商用のルータ、CADNETIX を用いて同じランダムネットを同じ配線条件で処理していただいた結果、約 10% 以上良い配線率が得られることが分かった<sup>1)</sup>。この理由として、1) スレーブの配線アルゴリズムの未熟なこと、2) 並列処理方式を持つ問題、の2点が考えられる。前者は現在までに開発されている配線アルゴリズムを参考にしてアルゴリズムを改善すれば解決されるであろう。一方後者に関しては、本処理方式の早い者勝ちの競合裁定方式が配線率の良くない原因の一つであると推測されるので、今後引き剥がし処理を取り入れた新しい並列処理方式を研究する必要がある。

**謝辞** 本研究に対して、実データを提供していただき、また商用ルータによる配線処理を行っていただいたイビデン(株)ソフト開発センターの足立和正次長と清水賢治氏、TOP-1 の使用に当たってご協力いただいた日本 IBM 東京基礎研究所の方々、および本研究の初期段階に参加された卒業生の佐々木茂高君(現日本電気)に対し厚く感謝いたします。

## 参 考 文 献

- 1) Takahashi, Y. and Sasaki, S.: Parallel Automated Wire-Routing with a Number of Competing Processors, *Proc. ACM International Conf. on Supercomputing*, pp. 310-317 (1990).
- 2) 高橋義造, 佐々木茂高: 競合プロセッサ群による配線問題の並列処理, 情報処理学会研究報告, Vol. 90, No. 29, 90-ARC-82 (1990).

3) Ohtsuki, T.: Maze-running and Line-search Algorithms, *Layout Design and Verification*, Ohtsuki, T. ed., Elsevier Science Publishers B. V. (1986).

4) 遠藤俊雄, 松尾賢二, 白方新洋, 槌谷 一, 高橋義造: 二進木マシン Coral 68 K のシステム構成と性能評価, 情報処理学会研究報告, Vol. 87, No. 45, 87-CA-66-5 (1987).

5) 高橋義造, 遠藤俊雄, 松尾賢二, 槌谷 一: 二進木並列計算機 Coral 68 K の開発と性能評価, 情報処理学会論文誌, Vol. 30, No. 1, pp. 46-57 (1989).

6) 清水茂則, 大庭伸之, 森脇 淳, 中田武男, 小原盛幹: 高性能マルチプロセッサ・ワークステーション TOP-1, 並列シンポジウム JSP '89, pp. 155-162 (1989).

7) 中田登志之, 田辺記生, 梶原信樹, 松下 智, 小野塚裕美, 浅野由裕, 小池誠彦: 並列シミュレーションマシン Cenju, 並列シンポジウム JSP '90, pp. 353-360 (1990).

8) 河村 薫, 進藤達也, 澁谷利行, 三渡秀樹, 大木由江, 土肥実久: 超並列配線マシン MAPLE-RP, 並列シンポジウム JSP '91, pp. 373-379 (1991).

9) 伊達 博, 大嶽能久, 瀧 和男: 並列オブジェクトモデルに基づく LSI 配線プログラム, 並列シンポジウム JSP '91, pp. 381-388 (1991).

(平成3年8月1日受付)

(平成3年12月9日採録)



佐野 雅彦 (学生会員)

昭和 42 年生。平成 2 年徳島大学工学部情報工学科卒業。現在、同大学院修士課程在学中。配線問題の並列処理方式の研究に従事。



高橋 義造 (正会員)

昭和 7 年生。昭和 30 年京都大学工学部電気工学科卒業。同年(株)東芝に入社し、アナログ計算機、計算機制御、計算機基本ソフト、各種オンラインシステム等の研究開発に従事。昭和 52 年より徳島大学工学部教授。現在、知能情報工学科において、計算機アーキテクチャと並列処理の応用に関する教育・研究にたずさわっている。著書「制御工学」「電子計算機演習」(以上朝倉書店)、「計算機方式」(コロナ社)、編著「並列処理機構」(丸善)等。工学博士。電子情報通信学会、ソフトウェア科学会、ACM、IEEE Computer Society 各会員。